Distributed Systems

COMP90015 2019 Semester 1
Tutorial 09

Things to cover

1. SSL Explanation and Code Demonstration
2. Command line Arguments Demonstration
3. Operating Systems Support Questions (If time permits)

SSL

Secure Socket Layer is a protocol that is intended to provide a flexible means for clients and server to communicate using a secure channel.

It prevents eavesdropping, tampering and message forgery.

To use SSL, you need a Certificate that will identify who you are.

How does it work?

SSL Protocol

1. Agree on the cipher and hash functions that both client and server support.
2. Server authenticates by sending a certificate.
   The certificate contains: server name, trusted certificate authority, server's public key.
3. The client confirms the validity of the certificate.
4. Shared session key for the secure connection is generated.
   Client can encrypt a random number with the server's public key and send it to the server. Only the server can decrypt the random number with its private key. The agreed on random number is used to generate a unique session key.

After the steps above, secured connection begins. Data is encrypted and decrypted with the session key until the connection closes.

SSL and Java

Java Secure Socket Extension (JSSE) provides a set of packages which enable secure Internet communications.

It implements a Java technology version of the SSL/TLS protocols.

- `SSLServerSocketFactory (Server)`
- `SSLServerSocket (Server)`
- `SSLSocket (Client and Server)`
- `SSLSocketFactory (Client)`

SSL and Java

- Build server certificate
  - Keytool (part of JSE).
  - Allows users to administer their own public/private key pairs and associated certificates.
  - Create keystore, generate a self-signed certificate.

  -
    ```
    keytool -alias <aliasName> -genkey -keyalg RSA -keystore
    <path/to/keystore/keystoreName> -storepass <password> -validity
    <days> -keysize <2048>
    ```
- Define keystore location in server:
  - `javax.net.ssl.keyStore=<path/to/keystore/SERVERkeystoreName>`
  - `javax.net.ssl.keyStorePassword=<password>`
- Define trusted certificate in client:
  - `javax.net.ssl.trustStore=<path/to/keystore/CLIENTkeystoreName>`

SSL and Java

- The server will export their certificate to the client

  -
    ```
    keytool -alias <aliasName> -export -file <certificateName.crt> -keystore <path/to/keystore/SERVERkeystoreName>
    ```
- The client will then import the certificate

  -
    ```
    keytool -keystore <path/to/keystore/CLIENTkeystoreName> -import -file <certificateName.crt>
    ```

Generate a CSR:

-
  ```
  keytool -certreq -alias <aliasName> -file <certificateName.csr> -keystore <path/to/keystore/keystoreName>
  ```

SSL Code Demo

Command-line Arguments Demo

Operating Systems Support Questions

1. Discuss the difference between a network operating system and a distributed operating system.

2. What are the core OS components?

3. What is the kernel?

4. What is Supervisor Mode vs User Mode?

5. Discuss the potential advantages and disadvantages of using of multiple threads compared to multiple processes.

1. Discuss the difference between a network operating system and a distributed operating system.

1. Discuss the difference between a network operating system and a distributed operating system.

- A *networked operating system* provides support for networking operations. The users are generally expected to make intelligent use of the network commands and operations that are provided. Each host remains autonomous in the sense that it can continue to operate when disconnected from the networking environment.

- A *distributed operating system* tries to abstract the network from the user and thereby remove the need for the user to specify how the networking commands and operations should be undertaken. This is sometimes referred to as providing a *single system image*. Each host may not have everything that would be required to operate on its own, when disconnected from the network.

1. Discuss the difference between a network operating system and a distributed operating system.

Network operating system

- Users retain autonomy in managing their own processing resources
- It does not manage processes across the nodes
- Provides support for networking operations

1. Discuss the difference between a network operation system and a distributed operating system.

Distributed operating system

- Users are never concerned with where their programs run, or the location of any resources
- Has control over all the nodes in the system, and it transparently locates new processes at whatever node suits its scheduling policies
- Each host may not have everything that would be required to operate on its own
- Single system image
- Not widely adopted

2. What are the core OS components?

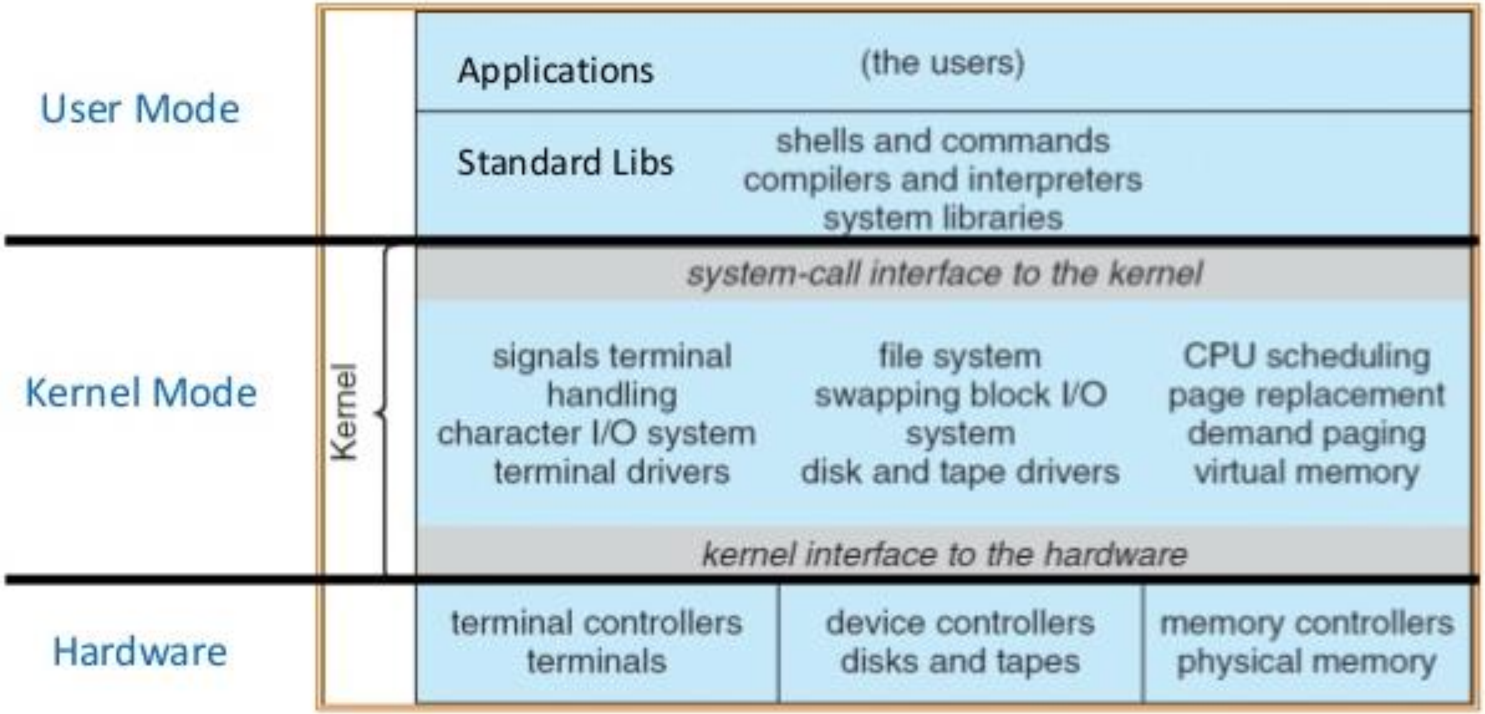
2. What are the core OS components?

- **Process manager** -- Handles the creation of processes, which is a unit of resource management, encapsulating the basic resources of memory (address space) and processor time (threads).
- **Thread manager** -- Handles the creation, synchronization and scheduling of one or more threads for each process. Threads can be scheduled to receive processor time.
- **Communication manager** -- Handles interprocess communication, i.e. between threads from different processes. In some cases this can be across different hosts

2. What are the core OS components?

- **Memory manager** -- Handles the allocation and access to physical and virtual memory. Provides translation from virtual to physical memory and handles paging of memory.
- **Supervisor** -- Handles privileged operations, i.e. those that directly affect shared resources on the host, e.g. to and from an I/O device. The supervisor is responsible for ensuring that host continues to provide proper service to each client.

A Real-World Example - UNIX



User Mode

Kernel Mode

Hardware

Kernel

Applications (the users)

Standard Libs

shells and commands
compilers and interpreters
system libraries

system-call interface to the kernel

signals terminal handling
character I/O system
terminal drivers

file system
swapping block I/O system
disk and tape drivers

CPU scheduling
page replacement
demand paging
virtual memory

kernel interface to the hardware

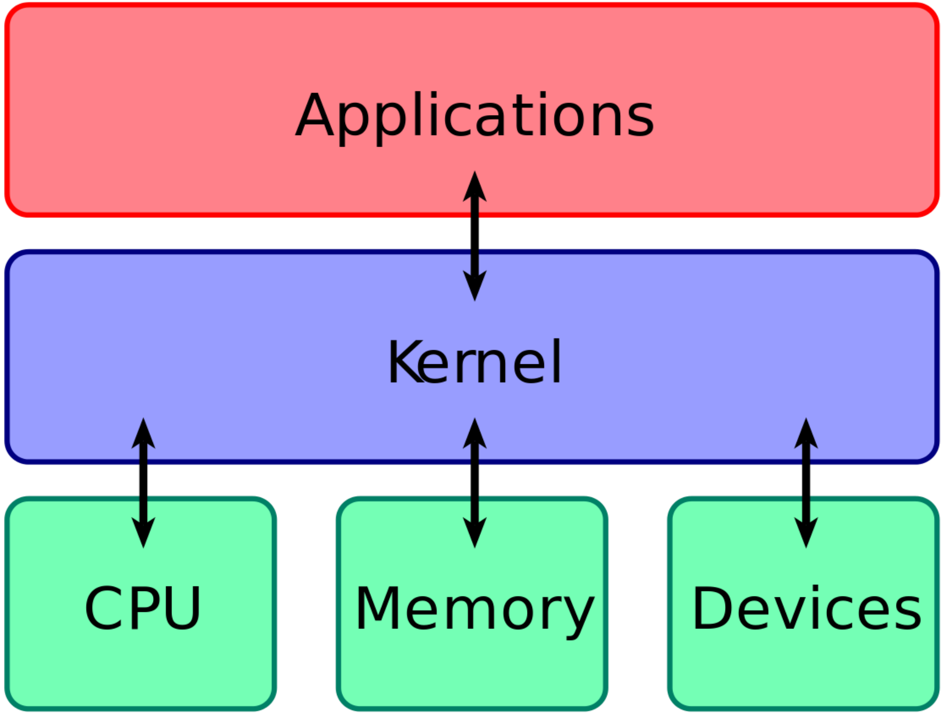
terminal controllers
terminals

device controllers
disks and tapes

memory controllers
physical memory

3. What is the kernel?

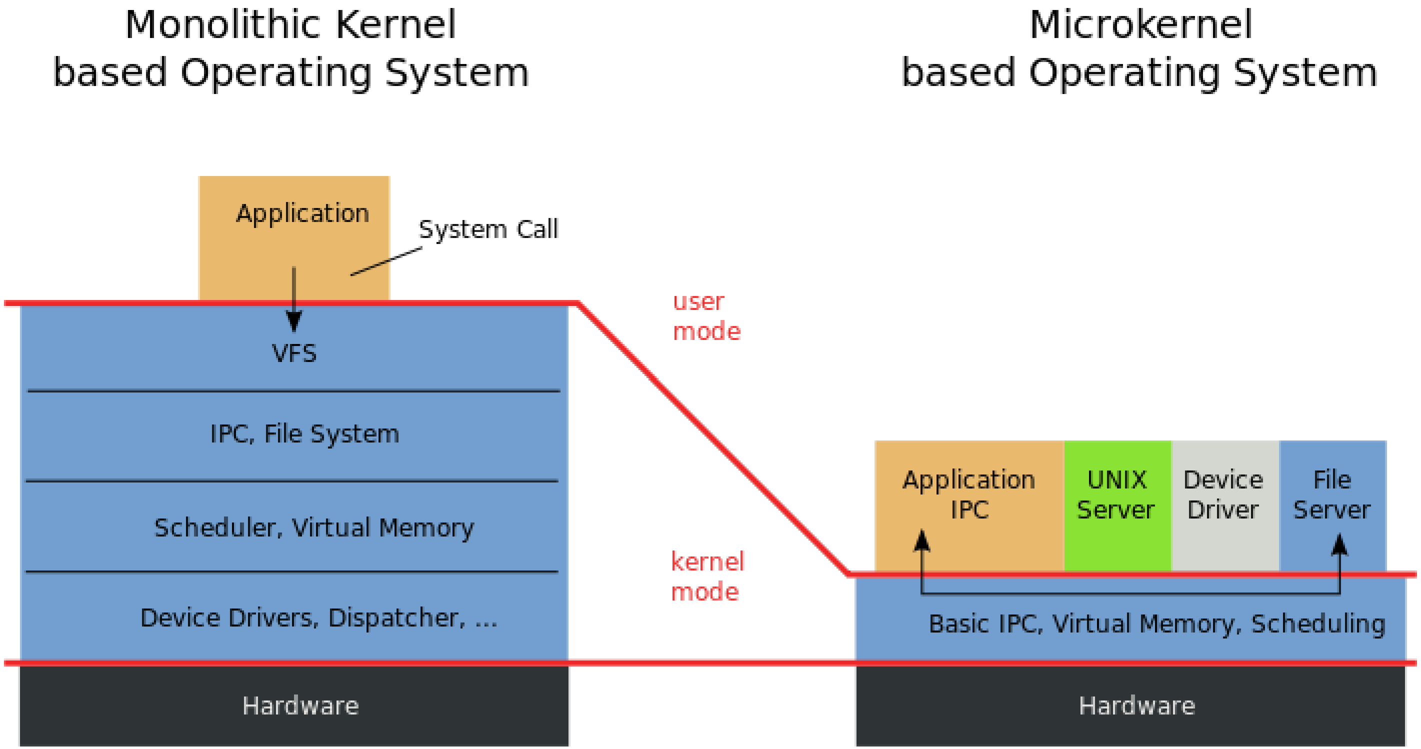
3. What is the kernel?



3. What is the kernel?

- Part of the Operating System
- Has full access to the host’s resources
- Kernel begins execution after the host is powered up and continues to execute while the host is operational
- The kernel has access to all resources and shares access to all other processes that executing on the host

Comparison of Kernel Implementations



4. What is Supervisor Mode vs User Mode?

4. What is Supervisor Mode vs User Mode?

- Operating modes supported by the hardware at the machine instruction level.
- **Supervisor / Kernel mode** -- instructions that execute while the processor is in supervisor (or privileged) mode are capable of accessing and controlling every resource on the host,
- **User mode** -- instructions that execute while the processor is in user (or unprivileged) mode are restricted, by the processor, to only those accesses defined or granted by the kernel.

5. Discuss the potential advantages and disadvantages of using of multiple threads compared to multiple processes.

5. Discuss the potential advantages and disadvantages of using of multiple threads compared to multiple processes.

Advantages

- Cheaper to allocate/deallocate
- Faster switching
- Easy to share resources via shared memory.

5. Discuss the potential advantages and disadvantages of using of multiple threads compared to multiple processes.

Disadvantages

- Synchronization overhead of shared data
- Shared process memory space
- Program debugging (synchronization, non-deterministic timing, accidental data corruption)